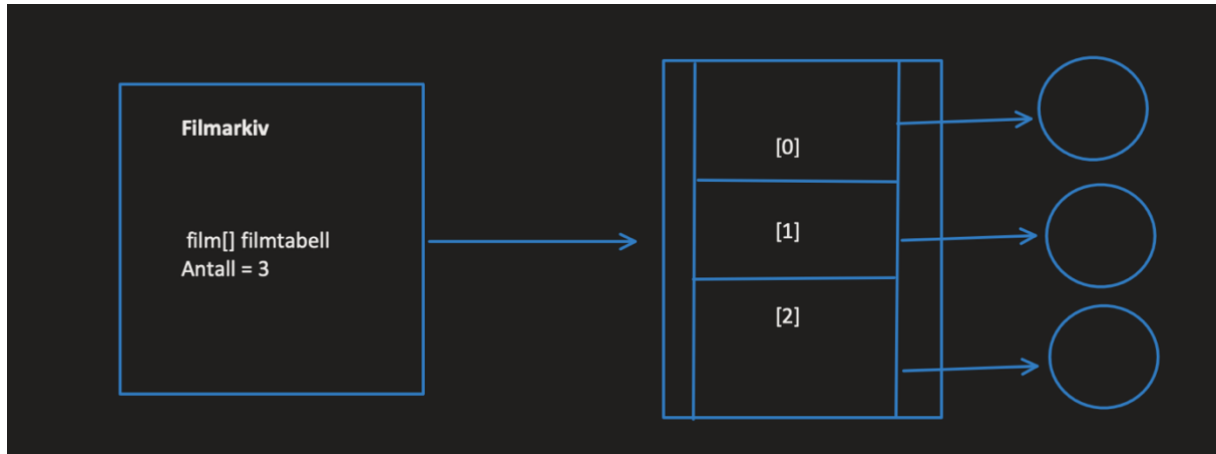
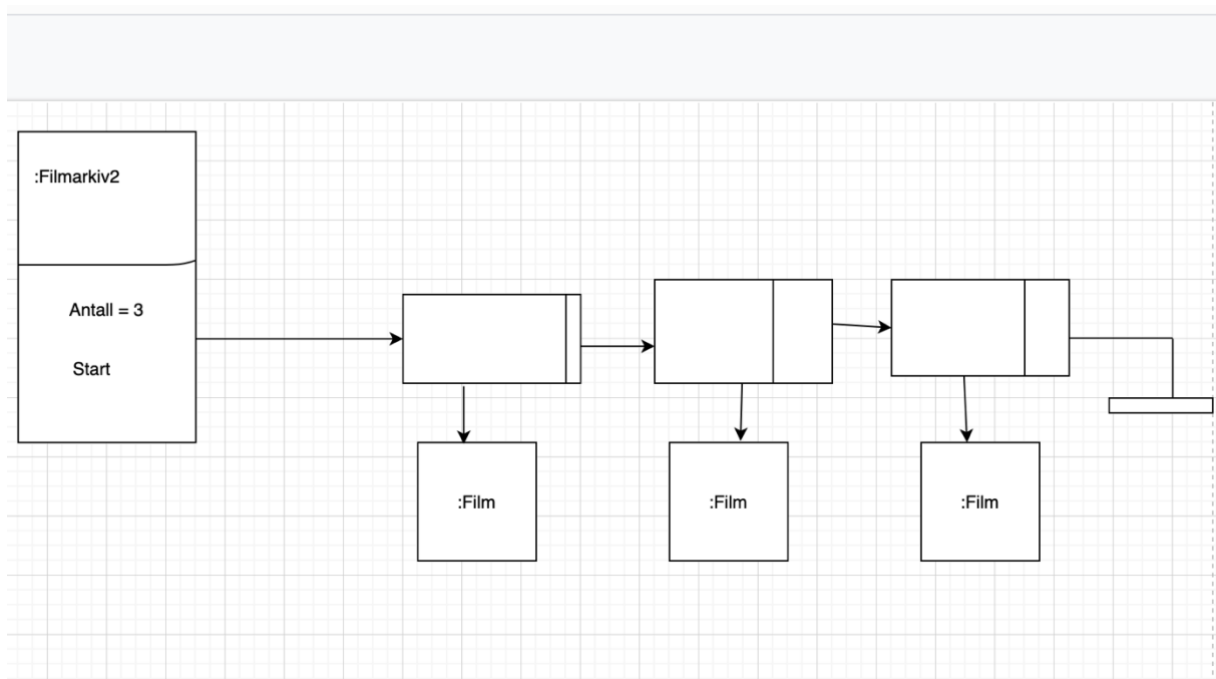


## Oblig 1 DAT102

### Objektdiagram – CD-arkiv



### Objektdiagram – CD-arkiv 2



### Oppgave 2a

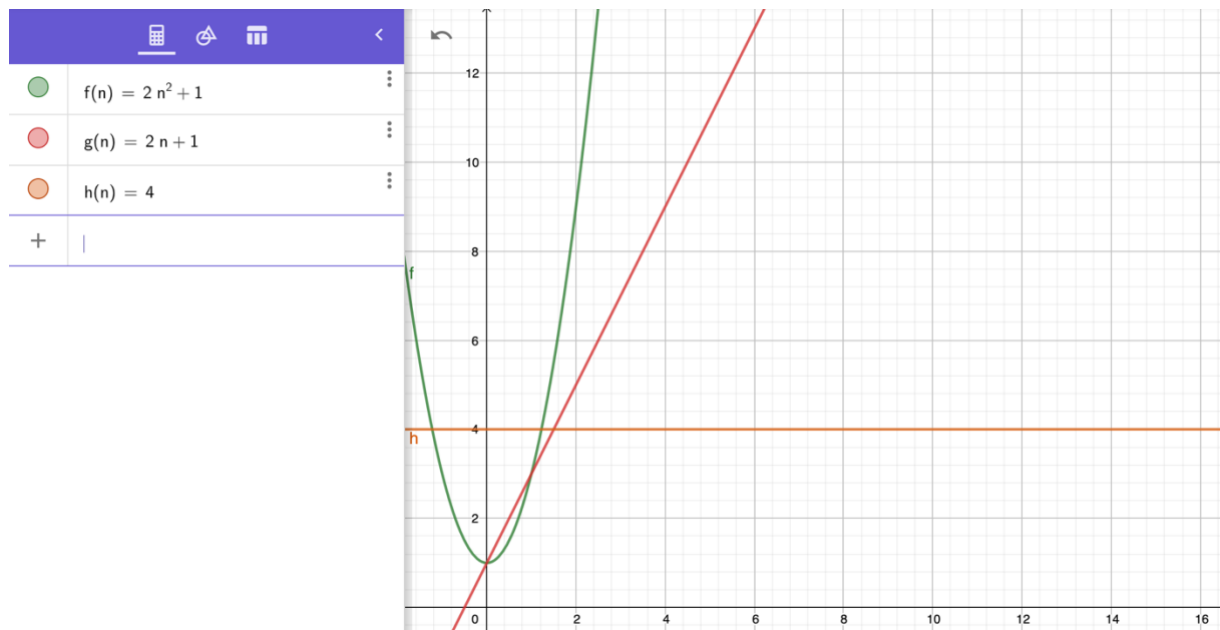
- i)  $t(n) = 4n^2 + 50n - 10 = O(n^2)$
- ii)  $O(n)$
- iii)  $O(n^3)$
- iv)  $O(\log_2 n)$

### Oppgave 2b.

$$O(n^2)$$

### Oppgave 2c.

	Algoritme A	Algoritme B	Algoritme C
Tilordning	$n + 1$	$n^2 + 1$	1
Addisjoner	$n + 1$	$n^2 + 1$	1
Multiplikasjoner	0	0	1
Divisjoner	0	0	1
Totalt antall operasjoner	$2n + 1$	$2n^2 + 1$	4



2d)

$t(n)$	$\frac{t(10^6)}{10^6}$
$\log_2 n$	$1,99 * 10^{-5s}$
$n$	1s

$n \log_2 n$	19,9s
$n^2$	11,6 dager
$n^3$	31710 år

2e)

Effektiviteten er:  $O(n^2)$

$$\text{Summen} = 1+2+3+ \dots + n-1 = \frac{n(n-1)}{2}$$

Oppgave 2f)

$$t1 = O(n^3)$$

$$t2 = O(\log_2 n)$$

$$t3 = O(n \log_2 n)$$

$$t4 = O(n)$$

Algoritmen som er mer aktiv er algoritme T2, dette er fordi den har vekstfunksjon som varer mest.

Algoritmen som er minst effektiv er algoritme t1, dette er fordi den har vekstfunksjonen som vokser raskest.

### Oppgave 3

Tidskompleksitet:

i)

antallSjanger(Sjanger sjanger)

O notasjon =  $O(n \cdot k)$

ii)

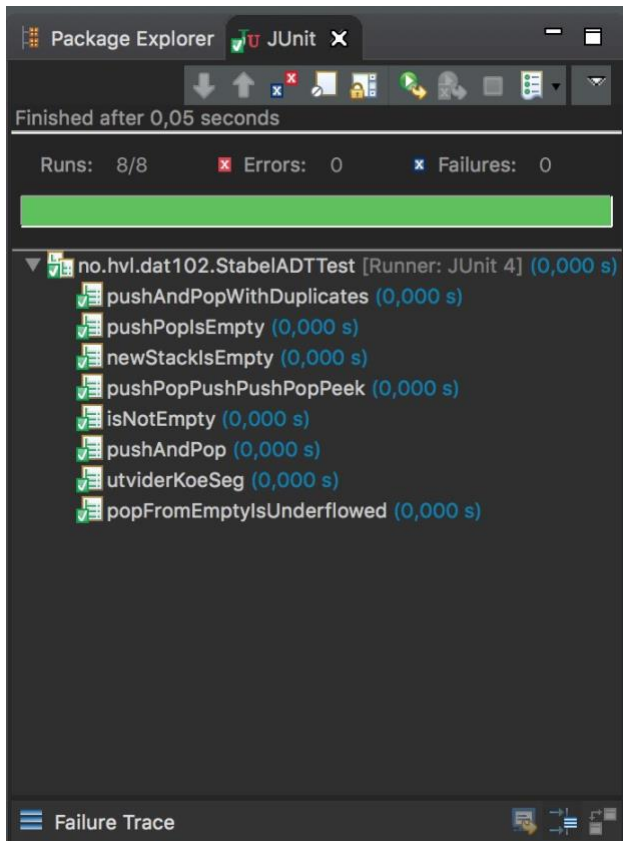
skrivUtStatistikk(FilmarkivADT film)

O notasjon =  $O(n^2)$

### Oppgave 4

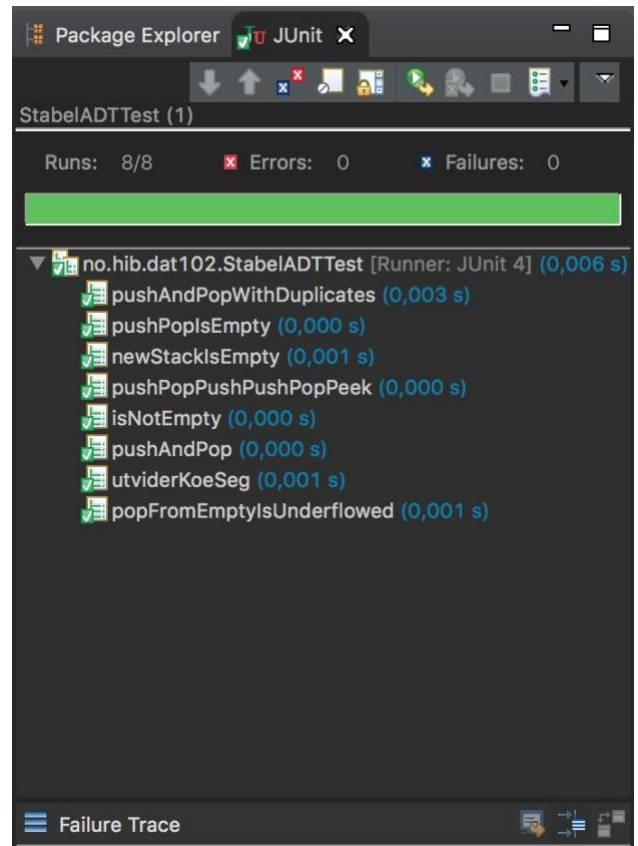
4a)

#### Stabel med unntak:



4b)

#### Stabel uten unntak:



i)

Public final void pushPopErTom():

- Setter først usikker kode i en try-kropp.
- Legger til et element(e0) til en tom stabel.
- Fjerner øverste element i stabel(e0).
- «asserttrue» returnerer true om parameteren er true. erTom() returnerer true hvis stabelen er tom. Asserttrue tester derfor om stabelen er tom eller ikke.
- Catch (EmptyCollectionException) fanger opp hvis pop metoden prøver å fjerne et element fra stabelen når den allerede er tom, og kaller på fail metoden som er en metode i Assert klassen som feiler testen og skriver ut feilmelding.

Testen går altså ut på å sjekke at hvis vi har en tom stabel og legger til element, og fjerner et element igjen så vil vi ende opp med en tom stabel igjen.

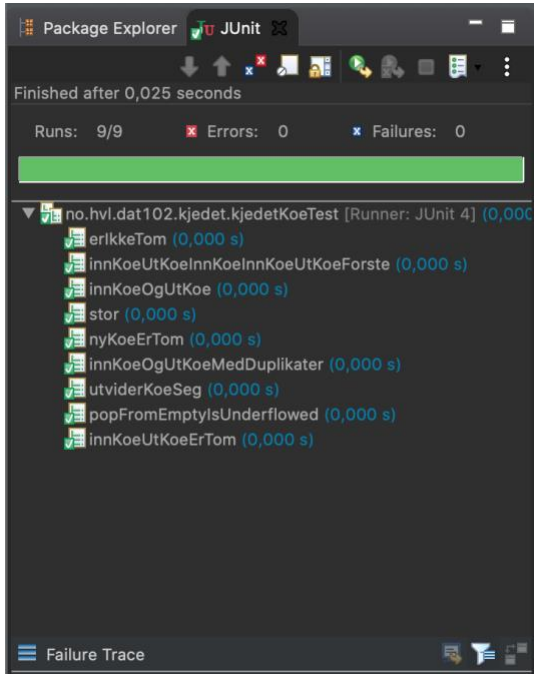
ii)

Public final void popFromEmptyIsUnderflowed() throws EmptyCollectionException: -

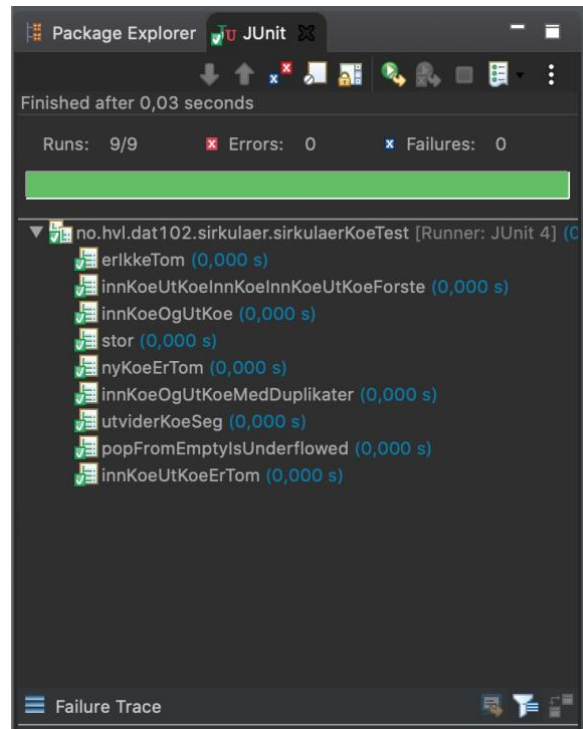
Expected = EmptyCollectionException.class betyr at det er forventet at testen skal kaste et unntak av denne klassen - Når stabel.pop() blir utført, prøver metoden å fjerne et element fra en tom stabel. Dette vil resultere i at et EmptyCollectionException unntak blir kastet. Så testen blir altså vellykket om stabel.pop() feiler

4c)

Kjedet kø test:



Sirkulær kø test:



## Oblig 1 rapport

**Gruppemedlemmer:** Christian Norill, Nicolai Gangdal og Thomas Vu

### Oppgavebeskrivelse:

I denne oppgaven (del 1) skal vi lage ett javaprogram som omhandler ett filmarkiv. Arkivet skal inneholde flere funksjoner for å legge til eller slette filmer søke etter tittel/produsent antall sjangere. En film inneholder sjanger, tittel på film, produsent, og år for lansering.

Ved å benytte oss av en abstrakt datatype (ADT) har vi muligheten til å lage en generell «oppskrift» på hva som skal skje i programmet.

Neste del av oppgaven (del 2) skulle vi utvide programmet ved hjelp av lineær kjedet struktur. Det vil si at vi brukte objekt referanser (som variabler) for å lenke et objekt til ett annet. Objekt variablene representerer en fysisk adresse i minnet for et objekt. Fordelene med denne strukturen er at Kjedet struktur er fleksibel og da slepper en og utvide arkivet for å legge til elementer, mens Oblig 1 må en oppdatere tabellen for å utvide. Fordelen med at vi brukte en ADT i oppgave 1 er at denne kunne gjenbrukes i del 2 av obligen.

### Plan for arbeidet, og hvordan har gruppearbeidet fungert:

Gruppearbeidet har fungert veldig bra. Vi har jobbet godt sammen og diskutert hvordan vi skal angripe de forskjellige arbeidsoppgavene. Det som er positivt med å jobbe i grupper er

at når en sitter fast og ikke komme noen vei, har vi alltid en eller to å diskutere problemet med, slik at vi kommer raskt frem til en god løsning.

På labben har vi jobbet samtidig med arbeidsoppgaven og diskutert eventuelle feil eller andre metoder å skrive koden på, og utbedret disse med å se på hverandre sine program for å finne den beste løsningen.

### Beskrivelse av arbeidsprosessen:

Arbeidsprosessen har gått ut på å stille på forelesningene, og deretter jobbet på labben med oppgavene både når vi har hatt labb og på fritiden.

### Avslutning, hva har vi lært:

I dette prosjektet har vi lært nytten av å utforme en generell ADT som kan brukes til flere klasser. Vi har og lært å bruke kjedet struktur, og fordelen med dette istedenfor tabeller. Vi har lært mer om enhetstesting, og hvordan å skrive egne tester. I tillegg til dette har vi lært mer om analyse, og viktigheten i at kode ikke bare skal kunne fungere, men også være mest mulig effektiv. Alt i alt synes vi prosjektet har vært en lærerik prosess.

### Klassekart:

